In the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (previously presented) A method for producing ephemeral, symmetric encryption keys at a first station for mutual authentication and secure distribution of a random session-specific symmetric encryption key in a communication session with a second station, comprising:

assigning a session key in the first station, in response to a request to initiate a communication session received by the first station during a session key initiation interval for use in a first exchange of a plurality of exchanges executed for distributing the symmetric encryption key produced for use in the communication session;

associating, in the first station, a set of intermediate data keys, different from said session key, with said request for use in said plurality of exchanges;

in the first exchange, sending at least one message carrying said session key to the second station, and receiving a response from the second station including a shared parameter, which is shared between the first station and the second station, or between the first station and a user at the second station, the shared parameter being encrypted using said session key to verify receipt of the session key by the second station and to identify the second station or the user of the second station; and

in another exchange in the plurality of exchanges, sending, after verifying in said first station receipt of the session key by the second station, at least one message carrying an encrypted version of one of the intermediate data keys from said set of intermediate data keys to be accepted as the symmetric encryption key for use by the first and second stations during the communication session.

2. (previously presented) The method of claim 1, including distributing symmetric encryption keys for use in a plurality of communication sessions using respective pluralities of exchanges, and using said session key for first exchanges in the respective pluralities of exchanges for initiating communication sessions in the plurality of communication sessions initiated with the first station, during said session key initiation interval, and using other session keys after expiry of said session key initiation interval.

3. (previously presented) The method of claim 2, including associating a unique set of intermediate data keys with each session key.

4. (previously presented) The method of claim 1, including:

providing a buffer at the first station;

storing an ephemeral set of session keys in the buffer for respective session key lifetimes;

associating respective session key initiation intervals with said session keys stored in said buffer;

using session keys from the set of session keys from said buffer as session keys in response to requests received by said first station during said respective, associated session key initiation intervals;

removing session keys from said buffer upon expiry of the respective session key lifetimes.

5. (canceled)

6. (previously presented) The method of claim 4, wherein the session key lifetimes have respective lengths longer or equal to a time required for the plurality of exchanges used to distribute the symmetric encryption key for use in a communication session can be completed in expected circumstances.

7. (previously presented) The method of claim 4, wherein the session key lifetimes have respective lengths which are a multiple M times a time required for the plurality of exchanges used to distribute the symmetric encryption key for use in a communication session can be completed in expected circumstances, where M is less than or equal to 10.

8. (previously presented) A data processing apparatus, comprising:

a processor associated with a first station, a communication interface adapted for connection to a communication medium, and memory storing instructions for execution by the data processor, the instructions including

logic to receive a request via the communication interface for initiation of a

communication session between a first station and a second station;

logic to provide symmetric encryption keys in response to a request received by said processor for initiation of a communication session between the first station and the second station, including logic to execute a plurality of exchanges to distribute the symmetric encryption key for use in the communication session, logic to provide a session key for use during a session key initiation interval, and to associate, in said first station, a set of intermediate data keys, different from said session key, with said request for use in said plurality of exchanges, and logic to send in a first exchange in said plurality of exchanges at least one message carrying said session key to the second station, and to receive a response from the second station including a shared parameter encrypted using said session key to verify receipt of the session key and to identify the second station or the user of the second station; and

logic to send, after verifying receipt of the session key at the second station, at least one message carrying, in another exchange in said plurality of exchanges, an encrypted version of one of said set of intermediate data keys to be accepted as the symmetric encryption key for use by the first and second stations during the communication session.

9. (previously presented) The apparatus of claim 8, including logic to distribute symmetric encryption keys for use in a plurality of communication sessions using respective pluralities of exchanges, and to use said session key for first exchanges in the respective pluralities of exchanges for distributing the symmetric encryption keys in the plurality of communication sessions initiated with the first station, during said session key initiation interval, and to use other session keys after expiry of said session key initiation interval.

10. (previously presented) The apparatus of claim 9, including logic to associate a unique set of intermediate data keys with each session key.

11. (previously presented) The apparatus of claim 8, including

a buffer at the first station;

logic to store a set of session keys in the buffer for respective session key lifetimes, to associate respective session key initiation intervals with particular session keys in said set of session keys stored in said buffer, to use session keys from said buffer as session keys in

response to requests received by said first station during said respective session key initiation intervals, and to remove session keys in said set of session keys from said buffer after expiry of the respective session key lifetimes.

12. (canceled).

13. (previously presented) The apparatus of claim 11, wherein the session key lifetimes have respective lengths longer or equal to a time required for the plurality of exchanges used to distribute the secret encryption key for use in a communication session can be completed in expected circumstances, and including logic to remove said session keys in said set of session keys from said buffer after expiry of the session key lifetimes.

14. (previously presented) The apparatus of claim 11, wherein the session key lifetimes have respective lengths which are a multiple M times a time required for the plurality of exchanges used to distribute the secret encryption key for use in a communication session can be completed in expected circumstances, and including logic to remove said session keys in said set of session keys from said buffer after expiry of the session key lifetimes.

15. (previously presented) An article, comprising:

machine readable data storage medium having computer program instructions stored therein for establishing a communication session on a communication medium between a first data processing station and a second data processing station having access to the communication medium, said instructions comprising

logic to receive a request via the communication interface for initiation of a communication session between a first station and a second station;

logic to provide symmetric encryption keys in response to a request received by said processor for initiation of a communication session between the first station and the second station, including logic to execute a plurality of exchanges to distribute the symmetric encryption key for use in the communication session, logic to provide a session key for use during a session key initiation interval, and to associate, in said first station, a set of intermediate data keys, different from said session key, with said request for use in said plurality of exchanges, and logic

to send in a first exchange in said plurality of exchanges at least one message carrying said session key to the second station, and to receive a response from the second station including a shared parameter encrypted using said session key to verify receipt of the session key and to identify the second station or the user of the second station; and

logic to send, after verifying receipt of the session key at the second station, at least one message carrying, in another exchange in said plurality of exchanges, an encrypted version of one of said set of intermediate data keys to be accepted as the symmetric encryption key for use by the first and second stations during the communication session.

16. (previously presented) The article of claim 15, wherein the instructions include logic to distribute secret encryption keys for use in a plurality of communication sessions using respective pluralities of exchanges, and to use said session key for first exchanges in the respective pluralities of exchanges for assigning secret encryption keys in the plurality of communication sessions initiated with the first station, during said session key initiation interval, and to use other session keys after expiry of said session key initiation interval.

17. (previously presented) The article of claim 16, wherein the instructions include logic to associate a unique set of ephemeral intermediate data keys with each session key.

18. (previously presented) The article of claim 15,

the first station includes a buffer; and

the instructions include logic to store a set of session keys in the buffer for respective session key lifetimes, to associate respective session key initiation intervals with particular session keys in said set of session keys stored in said buffer, to use session keys from said buffer as session keys in response to requests received by said first station during said respective session key initiation intervals, and to remove session keys in said set of session keys from said buffer after expiry of the respective session key lifetimes.

19. (canceled).

20. (previously presented) The article of claim 18, wherein the session key lifetimes have respective lengths longer or equal to a time required for the plurality of exchanges used to distribute the secret encryption key for use in a communication session can be completed in expected circumstances, and the instructions include logic to remove said session keys in said set of session keys from said buffer after expiry of the session key lifetimes.

21. (previously presented) The article of claim 18, wherein the session key lifetimes have respective lengths which are a multiple M times a time required for the plurality of exchanges used to distribute the secret encryption key for use in a communication session can be completed in expected circumstances, and the instructions include logic to remove said session keys in said set of session keys from said buffer after expiry of the session key lifetimes.

22. (previously presented) The method of claim 1, wherein the encrypted version of one of said set of intermediate data keys to be accepted as the symmetric encryption key is encrypted using a shared secret credential.

23. (currently amended) The method of claim 1, wherein the plurality of exchanges includes an iterative process including n iterations, in which for each iteration (i), the first station sends a message carrying intermediate data key (i) encrypted with intermediate data key (i-1), [[and]] the second station obtains intermediate key (i) by decrypting the message with intermediate key (i-1) and returns a message to the first station carrying a hashed version of the intermediate data key (i) encrypted using the intermediate data key (i), <u>and the first station decrypts the hashed version of intermediate data key (i) using intermediate data key (i), and determines that the hash version of intermediate data key (i) matches an expected version of by intermediate data key (i),</u> until the n-th iteration in which the first station sends intermediate data key (n) as the encrypted version of one of said set of intermediate data keys to be accepted as the symmetric encryption key, encrypted using a first shared secret credential, [[and]] the second station, after obtaining intermediate data key (n) by decrypting the message with the first shared secret credential, returns a message to the first station carrying a hashed version of intermediate data key (n) encrypted using the first shared secret credential, <u>and the first station decrypts the hashed version of intermediate data key (n) using the first shared secret credential, and determines that the</u>

hashed version of intermediate data key (n) matches an expected version of by intermediate data key (n), and in (n+1)-th iteration, the first station sends intermediate data key (n) encrypted using a second shared secret credential, and the second station, after obtaining intermediate data key (n) by decrypting the message with the second shared secret credential, returns a message to the first station carrying a hashed version of intermediate data key (n) encrypted using the second shared secret credential, and the first station decrypts the hashed version of intermediate data key (n) using the second shared secret credential, and determines that the hashed version of intermediate data key (n) matches an expected version of by intermediate data key (n).

24. (currently amended) The method of claim 1, wherein the plurality of exchanges includes an iterative process including n iterations, in which for each iteration (i), the first station sends a message carrying intermediate data key (i) encrypted with intermediate data key (i-1), [[and]] the second station obtains intermediate data key (i) by decrypting the message with intermediate data key (i-1), and returns a message to the first station carrying a hashed version of the intermediate data key (i) encrypted using the intermediate data key (i), and the first station decrypts the hashed version of intermediate data key (i) using intermediate data key (i), and determines that the hash version of intermediate data key (i) matches an expected version of by intermediate data key (i), until the n-th iteration in which the first station sends intermediate data key (n) as the encrypted version of one of said set of intermediate data keys to be accepted as the symmetric encryption key, encrypted using a first shared secret credential and intermediate data key (n-1), and the second station, after obtaining intermediate data key (n) by decrypting the message with the first shared secret credential and intermediate data key (n-1), returns a message to the first station carrying a hashed version of intermediate data key (n) encrypted using the first shared secret credential and intermediate data key (n), and the first station decrypts the hashed version of intermediate data key (n) using the first shared secret credential and intermediate data key (n), and determines that the hashed version of intermediate data key (n) matches an expected version of by intermediate data key (n), and in (n+1)-th iteration the first station sends intermediate data key (n) encrypted using a second shared secret credential and intermediate data key (n), and the second station after obtaining intermediate data key (n) by decrypting the message with the second shared secret credential and intermediate data key (n), returns a message to the first station carrying a hashed version of intermediate data key (n) encrypted using the second shared

secret credential and intermediate key (n) , and the first station decrypts the hashed version of intermediate data key (n) using the second shared secret credential and intermediate data key (n), and determines that the hashed version of intermediate data key (n) matches an expected version of by intermediate data key (n) .

25. (previously presented) The apparatus of claim 8, wherein the encrypted version of one of said set of intermediate data keys to be accepted as the symmetric encryption key is encrypted using a shared secret credential.

26. (currently amended) The apparatus of claim 8, wherein the plurality of exchanges includes an iterative process including n iterations, in which for each iteration (i), the first station sends a message carrying intermediate data key (i) encrypted with intermediate data key (i-1), [[and]] the second station obtains intermediate key (i) by decrypting the message with intermediate key (i-1) and returns a message to the first station carrying a hashed version of the intermediate data key (i) encrypted using the intermediate data key (i), and the first station decrypts the hashed version of intermediate data key (i) using intermediate data key (i), and determines that the hash version of intermediate data key (i) matches an expected version of by intermediate data key (i), until the n-th iteration in which the first station sends intermediate data key (n) as the encrypted version of one of said set of intermediate data keys to be accepted as the symmetric encryption key, encrypted using a first shared secret credential, and the second station, after obtaining intermediate data key (n) by decrypting the message with the first shared secret credential, returns a message to the first station carrying a hashed version of intermediate data key (n) encrypted using the first shared secret credential, and the first station decrypts the hashed version of intermediate data key (n) using the first shared secret credential, and determines that the hashed version of intermediate data key (n) matches an expected version of by intermediate data key (n), and in (n+1)-th iteration, the first station sends intermediate data key (n) encrypted using a second shared secret credential, and the second station, after obtaining intermediate data key (n) by decrypting the message with the second shared secret credential, returns a message to the first station carrying a hashed version of intermediate data key (n) encrypted using the second shared secret credential, and the first station decrypts the hashed version of intermediate data key

(n) using the second shared secret credential, and determines that the hashed version of intermediate data key (n) matches an expected version of by intermediate data key (n).

27. (currently amended) The apparatus of claim 8, wherein the plurality of exchanges includes an iterative process including n iterations, in which for each iteration (i), the first station sends a message carrying intermediate data key (i) encrypted with intermediate data key (i-1), [[and]] the second station obtains intermediate data key (i) by decrypting the message with intermediate data key (i-1), and returns a message to the first station carrying a hashed version of the intermediate data key (i) encrypted using the intermediate data key (i), and the first station decrypts the hashed version of intermediate data key (i) using intermediate data key (i), and determines that the hash version of intermediate data key (i) matches an expected version of by intermediate data key (i), until the n-th iteration in which the first station sends intermediate data key (n) as the encrypted version of one of said set of intermediate data keys to be accepted as the symmetric encryption key, encrypted using a first shared secret credential and intermediate data key (n-1), and the second station, after obtaining intermediate data key (n) by decrypting the message with the first shared secret credential and intermediate data key (n-1), returns a message to the first station carrying a hashed version of intermediate data key (n) encrypted using the first shared secret credential and intermediate data key (n), and the first station decrypts the hashed version of intermediate data key (n) using the first shared secret credential and intermediate data key (n), and determines that the hashed version of intermediate data key (n) matches an expected version of by intermediate data key (n), and in (n+1)-th iteration the first station sends intermediate data key (n) encrypted using a second shared secret credential and intermediate data key (n), and the second station after obtaining intermediate data key (n) by decrypting the message with the second shared secret credential and intermediate data key (n), returns a message to the first station carrying a hashed version of intermediate data key (n) encrypted using the second shared secret credential and intermediate key (n), and the first station decrypts the hashed version of intermediate data key (n) using the second shared secret credential and intermediate data key (n), and determines that the hashed version of intermediate data key (n) matches an expected version of by intermediate data key (n).

28. (previously presented) The article of claim 15, wherein the encrypted version of one of said set of intermediate data keys to be accepted as the symmetric encryption key is encrypted using a shared secret password.

29. (currently amended) The article of claim 15, wherein the plurality of exchanges includes an iterative process including n iterations, in which for each iteration (i), the first station sends a message carrying intermediate data key (i) encrypted with intermediate data key (i-1), [[and]] the second station obtains intermediate key (i) by decrypting the message with intermediate key (i-1) and returns a message to the first station carrying a hashed version of the intermediate data key (i) encrypted using the intermediate data key (i), and the first station decrypts the hashed version of intermediate data key (i) using intermediate data key (i), and determines that the hash version of intermediate data key (i) matches an expected version of by intermediate data key (i), until the n-th iteration in which the first station sends intermediate data key (n) as the encrypted version of one of said set of intermediate data keys to be accepted as the symmetric encryption key, encrypted using a first shared secret credential, and the second station, after obtaining intermediate data key (n) by decrypting the message with the first shared secret credential, returns a message to the first station carrying a hashed version of intermediate data key (n) encrypted using the first shared secret credential, and the first station decrypts the hashed version of intermediate data key (n) using the first shared secret credential, and determines that the hashed version of intermediate data key (n) matches an expected version of by intermediate data key (n),and in (n+1)-th iteration, the first station sends intermediate data key (n) encrypted using a second shared secret credential, and the second station, after obtaining intermediate data key (n) by decrypting the message with the second shared secret credential, returns a message to the first station carrying a hashed version of intermediate data key (n) encrypted using the second shared secret credential, and the first station decrypts the hashed version of intermediate data key (n) using the second shared secret credential, and determines that the hashed version of intermediate data key (n) matches an expected version of by intermediate data key (n).

30. (currently amended) The article of claim 15, wherein the plurality of exchanges includes an iterative process including n iterations, in which for each iteration (i), the first station sends a message carrying intermediate data key (i) encrypted with intermediate data key (i-1), [[and]] the second station

second station obtains intermediate data key (i) by decrypting the message with intermediate data key (i-1), and returns a message to the first station carrying a hashed version of the intermediate data key (i) encrypted using the intermediate data key (i), and the first station decrypts the hashed version of intermediate data key (i) using intermediate data key (i), and determines that the hash version of intermediate data key (i) matches an expected version of by intermediate data key (i), until the n-th iteration in which the first station sends intermediate data key (n) as the encrypted version of one of said set of intermediate data keys to be accepted as the symmetric encryption key, encrypted using a first shared secret credential and intermediate data key (n-1), and the second station, after obtaining intermediate data key (n) by decrypting the message with the first shared secret credential and intermediate data key (n-1), returns a message to the first station carrying a hashed version of intermediate data key (n) encrypted using the first shared secret credential and intermediate data key (n), and the first station decrypts the hashed version of intermediate data key (n) using the first shared secret credential and intermediate data key (n), and determines that the hashed version of intermediate data key (n) matches an expected version of by intermediate data key (n), and in (n+1)-th iteration the first station sends intermediate data key (n) encrypted using a second shared secret credential and intermediate data key (n), and the second station after obtaining intermediate data key (n) by decrypting the message with the second shared secret credential and intermediate data key (n), returns a message to the first station carrying a hashed version of intermediate data key (n) encrypted using the second shared secret credential and intermediate key (n), and the first station decrypts the hashed version of intermediate data key (n) using the second shared secret credential and intermediate data key (n), and determines that the hashed version of intermediate data key (n) matches an expected version of by intermediate data key (n).

///